# Project 4 :Hotel Management System

Design an Entity-Relationship schema for a multi-city hotel chain management system. The database must maintain hotels identified by hotel code, name, city, manager, number of rooms, and star rating. Rooms have room number, type, price per night, availability status, and belong to a hotel.

Guests have guest ID, name, loyalty level, booking history, and feedback given for bookings. Bookings have booking ID, guest, room, check-in and check-out dates, and total bill. Employees have employee ID, name, role, hotel assigned, and shift details.

Each hotel has multiple rooms and employees, and is managed by a manager who is also an employee. Guests can book rooms in any hotel and can have multiple active or past bookings. Rooms can be booked by different guests over time, but only one guest can occupy a room at a given time.

Employees are assigned to a specific hotel and can work in different shifts and roles. Loyalty level of a guest is updated based on their booking history and feedback. Feedback is linked to specific bookings and can influence loyalty level updates.
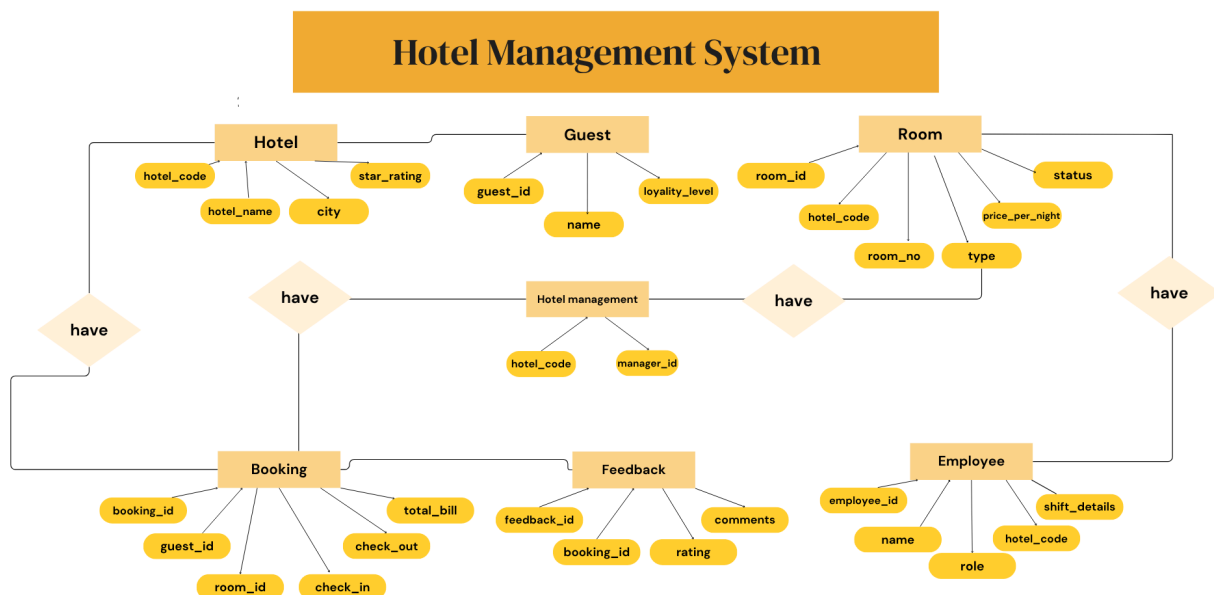
🧾 SQL Table Creation Statements:

-- Hotel table

```
CREATE TABLE Hotel (
    hotel_code INT PRIMARY KEY,
    name VARCHAR(100),
    city VARCHAR(100),
    star_rating INT
);
```

-- Room table
```
CREATE TABLE Room (
    room_id INT PRIMARY KEY AUTO_INCREMENT,
    hotel_code INT,
    room_number VARCHAR(10),
    type VARCHAR(50),
    price_per_night DECIMAL(10,2),
    availability_status VARCHAR(20),
    FOREIGN KEY (hotel_code) REFERENCES Hotel(hotel_code)
);
```

-- Guest table
```
CREATE TABLE Guest (
    guest_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    loyalty_level VARCHAR(20)
);
```

-- Booking table
```
CREATE TABLE Booking (
    booking_id INT PRIMARY KEY AUTO_INCREMENT,
    guest_id INT,
    room_id INT,
    check_in DATE,
    check_out DATE,
    total_bill DECIMAL(10,2),
    FOREIGN KEY (guest_id) REFERENCES Guest(guest_id),
    FOREIGN KEY (room_id) REFERENCES Room(room_id)
);
```

-- Feedback table
```
CREATE TABLE Feedback (
```

```sql
    feedback_id INT PRIMARY KEY AUTO_INCREMENT,
    booking_id INT,
    rating INT CHECK (rating BETWEEN 1 AND 5),
    comments TEXT,
    FOREIGN KEY (booking_id) REFERENCES Booking(booking_id)
);
```

-- Employee table
```sql
CREATE TABLE Employee (
    employee_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    role VARCHAR(50),
    hotel_code INT,
    shift_details VARCHAR(50),
    FOREIGN KEY (hotel_code) REFERENCES Hotel(hotel_code)
);
```

-- Manager Assignment (1 manager per hotel, and manager is an employee)
```sql
CREATE TABLE HotelManager (
    hotel_code INT PRIMARY KEY,
    manager_id INT,
    FOREIGN KEY (hotel_code) REFERENCES Hotel(hotel_code),
    FOREIGN KEY (manager_id) REFERENCES Employee(employee_id)
);
```

/* ========== Insert the values ========== */

```sql
INSERT INTO Hotel (hotel_code, name, city, star_rating) VALUES
(101, 'Sunrise Residency', 'Mumbai', 4),
(102, 'Ocean Breeze', 'Chennai', 5),
(103, 'Hilltop Retreat', 'Shimla', 3);

INSERT INTO Room (hotel_code, room_number, type, price_per_night,
availability_status) VALUES
(101, '101A', 'Deluxe', 4500.00, 'Available'),
(101, '102B', 'Suite', 6500.00, 'Occupied'),
(102, '201C', 'Standard', 3500.00, 'Available'),
(103, '301D', 'Deluxe', 4000.00, 'Under Maintenance');
```

```sql
INSERT INTO Guest (name, loyalty_level) VALUES
('Aishwarya Iyer', 'Gold'),
('Rithika Menon', 'Silver'),
('Narashima Rao', 'Platinum');

-- Assuming room_id: 1 (101A), 2 (102B), 3 (201C), 4 (301D)
-- guest_id: 1 (Aishwarya), 2 (Rithika), 3 (Narashima)
INSERT INTO Booking (guest_id, room_id, check_in, check_out, total_bill)
VALUES
(1, 1, '2025-06-10', '2025-06-12', 9000.00),
(2, 2, '2025-06-08', '2025-06-11', 19500.00),
(3, 3, '2025-06-13', '2025-06-15', 7000.00);

-- booking_id: 1 (Aishwarya), 2 (Rithika), 3 (Narashima)
INSERT INTO Feedback (booking_id, rating, comments) VALUES
(1, 5, 'Excellent stay, very clean and comfortable!'),
(2, 4, 'Great service but Wi-Fi was slow.'),
(3, 3, 'Average experience, food could be better.');

INSERT INTO Employee (name, role, hotel_code, shift_details) VALUES
('Sahithya Reddy', 'Manager', 101, 'Morning'),
('Udayasri Nair', 'Receptionist', 101, 'Evening'),
('Arjun Singh', 'Manager', 102, 'Morning'),
('Meera Sharma', 'Housekeeping', 102, 'Night'),
('Rajesh Khanna', 'Manager', 103, 'Day'),
('Kavita Das', 'Chef', 103, 'Morning');

-- Match manager_id to employee_id from Employee table
-- Assuming Sahithya = 1, Arjun = 3, Rajesh = 5
INSERT INTO HotelManager (hotel_code, manager_id) VALUES
(101, 1),
(102, 3),
(103, 5);
```

room
- 🔑 room_id INT
- ◇ hotel_code INT
- ◇ room_number VARCHAR(10)
- ◇ type VARCHAR(50)
- ◇ price_per_night DECIMAL(10,2)
- ◇ availability_status VARCHAR(2...
- Indexes ▶

hotelmanag... ▼
- 🔑 hotel_code INT
- ◇ manager_id INT
- Indexes ▶

booking
- 🔑 booking_id INT
- ◇ guest_id INT
- ◇ room_id INT
- ◇ check_in DATE
- ◇ check_out DATE
- ◇ total_bill DECIMAL(10,...
- Indexes ▶

employee ▼
- 🔑 employee_id INT
- ◇ name VARCHAR(100)
- ◇ role VARCHAR(50)
- ◇ hotel_code INT
- ◇ shift_details VARCHAR(5...
- Indexes ▶

hotel ▼
- 🔑 hotel_code INT
- ◇ name VARCHAR(100)
- ◇ city VARCHAR(100)
- ◇ star_rating INT
- Indexes ▶

feedba... ▼
- 🔑 feedback_id INT
- ◇ booking_id INT
- ◇ rating INT
- ◇ comments TEXT
- Indexes ▶

guest ▼
- 🔑 guest_id INT
- ◇ name VARCHAR(100)
- ◇ loyalty_level VARCHAR(2...
- Indexes ▶